WE CLAIM:

1. A method for protecting an operating system, comprising:

determining integrity data associated with an operating system binary, wherein the integrity data enables detection of a modification to the operating system binary; and

modifying a kernel with the integrity data, wherein the kernel is operable to employ the integrity data to detect the modification to the operating system binary.

2. The method of claim 1, wherein the integrity data further comprises at least one of a digital signature, and a hash associated with the operating system binary.

3. The method of claim 2, wherein the hash further comprises at least one a message digest, and a Secure Hash Algorithm (SHA).

4. The method of claim 1, wherein the modifying the kernel further comprises:

storing the integrity data in a data store; and

embedding the data store into the kernel.

5. The method of claim 4, wherein embedding the data store in the kernel further comprises at least one of digitally signing the data store, and encrypting the data store.

6. The method of claim 1, further comprising generating an operating system image based in part on the modified kernel and the operating system user level binary, wherein the operating system image comprises at least one of creating an archive file, a compressed file, and a Cabinet (CAB) file.

7. The method of claim 1, wherein the operating system binary further comprises at least one of an OS user level binary, and the kernel.

8. A method for protecting an operating system, comprising;

generating a first integrity data associated with an operating system binary;

modifying an operating system kernel with the first integrity data;

receiving a request associated with the operating system binary;

retrieving the first integrity data associated with the operating system binary;

determining if the first integrity data indicates tampering of the operating system binary; and

performing a tamper detection action if the first integrity data indicates tampering of the operating system binary.

9. The method of claim 8, wherein receiving the request further comprises receiving at least one of a read action, an execute operation, and an install request.

10. The method of claim 8, wherein performing the tamper detection action further comprises at least one of providing a tamper detection message, and quarantining the operating system binary.

11. The method of claim 8, wherein the first integrity data further comprises at least one of a digital signature, and a hash associated with the operating system binary.

12. The method of claim 11, wherein the hash further comprises at least one a message digest, and a Secure Hash Algorithm (SHA).

13. The method of claim 8, wherein modifying the operating system kernel with the first integrity data further comprises storing the first integrity data in at least one of a database, a file, and a program.

14. The method of claim 8, wherein modifying the operating system kernel further comprises associating the first integrity data with the operating system kernel.

15. The method of claim 14, where associating the first integrity data with the operating system kernel further comprises digitally signing the first integrity data with a digital key associated with the operating system kernel.

16. The method of claim 8, wherein determining if the first integrity data indicates tampering of the operating system binary further comprises:

determining a second integrity data associated with the operating system binary;

determining if the first integrity data is substantially different from the second integrity data; and

indicating tampering of the operating system binary if the first integrity data is substantially different from the second integrity data.

17. The method of claim 16, wherein determining if the first integrity data is substantially different from the second integrity data further comprises comparing the second integrity data to the first integrity data.

18. A method for protecting an operating system, comprising:

receiving a request associated with an operating system binary;

retrieving integrity data associated with the operating system binary; and

performing a tamper detection action if the integrity data indicates tampering of the operating system binary.

19. The method of claim 18, wherein receiving the request further comprises receiving at least one of a read action, an execute operation, and an install request.

20. The method of claim 18, wherein performing the tamper detection action further comprises at least one of providing a tamper detection message, and quarantining the operating system binary.

21. The method of claim 18, wherein determining if the integrity data indicates tampering of the operating system binary further comprises:

determining another integrity data associated with the operating system binary;

determining if the other integrity data is substantially different from the retrieved integrity data; and

indicating tampering of the operating system binary if the other integrity data is substantially different from the retrieved integrity data.

22. A computer-readable medium having computer-executable components for protecting an operating system, comprising:

a data store configured to receive and store a first integrity data, wherein the first integrity data is associated with an operating system binary; and

a tamper detection component, coupled to the data store, that is arranged to perform actions, including:

receiving a request to examine an operating system binary;

retrieving the first integrity data associated with the operating system binary;

determining if the first integrity data indicates tampering of the operating system binary; and

performing a tamper detection action if the first integrity data indicates tampering of the operating system binary.

23. The computer-readable medium of claim 22, wherein the computer-executable components are associated with an operating system kernel.

24. The computer-readable medium of claim 22, wherein performing the tamper detection action further comprises at least one of providing a tamper detection message, and quarantining the operating system binary.

25. The computer-readable medium of claim 22,wherein the first integrity data further comprises at least one of a digital signature, and a hash associated with the operating system binary.

26. The computer-readable medium of claim 22, wherein the operating system binary further comprises at least one of an OS user level binary, and a kernel.

27. The computer-readable medium of claim 22, wherein determining if the first integrity data indicates tampering of the operating system binary further comprises:

determining a second integrity data associated with the operating system binary;

determining if the first integrity data is substantially different from the second integrity data, and

indicating tampering of the operating system binary if the first integrity data is substantially different from the second integrity data.

28. The computer-readable medium of claim 22, wherein the second integrity data further comprises at least one of a digital signature, and a hash associated with the operating system binary.

29. An apparatus for protecting an operating system, comprising:

means for receiving a request to examine an operating system binary;

means for retrieving a first integrity data associated with the operating system binary;

means for determining a second integrity data associated with the operating system binary; and means for determining if the first integrity data is substantially different from the second integrity data, and if the first integrity data is substantially different from the second integrity data, a means for performing a tamper detection action.